Journal of Engineering and Applied Sciences 2 (9): 1387-1393, 2007
ISSN: 1816-949X

Particle Swarm Based PI Controller for Permanent Magnet Synchronous Machine

Lamia Benameur, Jihane Alami, Azzouz Loukdache and Abdelhakim El Imrani
Laboratoire Conception and Systèmes, Faculté des Sciences,
Université Mohammed V-Agdal, Rabat, Maroc

Abstract: Permanent magnet synchronous motors are often used in electrical drives because of their simple structures, ease of maintenance and high efficiency. However, these motors have a nonlinear characteristic arisen from motor dynamics and load characteristics. To overcome this problem, a new speed controller based on particle swarm optimisation for the drive system is proposed in this study. To illustrate the performance of the proposed controller, a conventional proportional integral controller is used too to the speed control of permanent magnet synchronous motors. Simulations are realized by the proposed strategy and the results are compared with the conventional control.

Key words: Particle swarm optimization, permanent magnet synchronous motors, optimal control, adaptive control

INTRODUCTION

Permanent Magnet Synchronous Motors (PMSM) are of great interest especially for industrial applications in low-medium power range, since it has superior features such as compact size, high torque/weight ratio, high torque/inertia ratio and absence of rotor losses (Slemon, 1994). However, the performance of the PMSM is very sensitive to parameter variations and is spoiled due to external load disturbances in the system. The conventional controller design, i.e., Proportional-Integrator, is based on mathematical model of the plant, which may often be unknown, less defined, nonlinear, complex and multivariable with parameter variation. Thus, the conventional PI controller is not an all-purpose solution for any motor drive applications.

To overcome these problems, several control strategies such as fuzzy logic control (Akcayol *et al*., 2002), sliding mode control (Karunadasa and Renfrew, 1991) and artificial neural network (Rahman and Hoque, 1998) have been proposed for speed and position control of PMSM. Recent literature has also explored the potentials of the Genetic Algorithm (GA) for motor drive applications (Karunadasa and Renfrew, 1991; Rahman and Hoque, 1998; Lee, 1990; Lin and Lee, 1991).

As an intelligent control technology, the PSO can give robust adaptive response of a drive with nonlinearity, parameter variation and load disturbance effect; that an exact mathematical model of system cannot be obtained at all (Lee, 1990).

In this study, a new Proportional-Integrator corrector based on Particle Swarm Optimisation (PIPSO) is designed for speed control of PMSM. In fact, the PIPSO speed controller is applied to the speed loop by replacing the conventional PI speed controller. An automatic tuning process using PSO is used to optimize the conventional PI parameters. Only two parameters are sought but tuning is complicated by a significant nonlinearity caused by saturation of the speed controller. This means that optimum controller settings depend on the form of the required speed demand.

MODELING OF PMSM DRIVE SYSTEM

The configuration of PMSM drive system is given in Fig. 1. The drive system is composed of speed controller (PIPSO, PIGA or conventional PI), a current regulator, a hysteresis band current controller, a three phase PWM inverter and a position encoder.

Figure 2 presents an equivalent circuit of PMSM and the 3 phase inverter.

Where, θ_r is the rotor position, ω_r the actual speed, i_a^*, i_b^*, i_c^* the reference phase currents and e_W represents the speed error. e_W is the difference between reference speed ω_r^* and actual speed ω_r. Using the speed error e_W, the speed controller generates I^* called reference current or control current.

Corresponding Author: Lamia Benameur, Laboratoire Conception and Systèmes, Faculté des Sciences, Université Mohammed V-Agdal, Rabat, Maroc

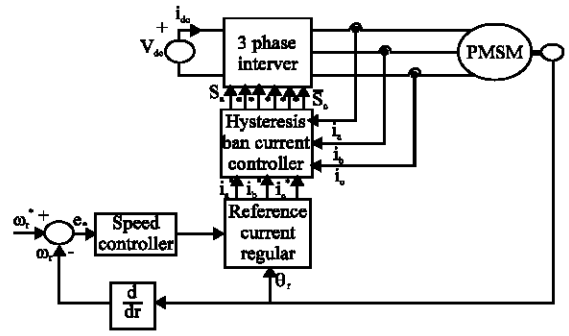

Fig. 1: Block diagram of PMSM speed drive system

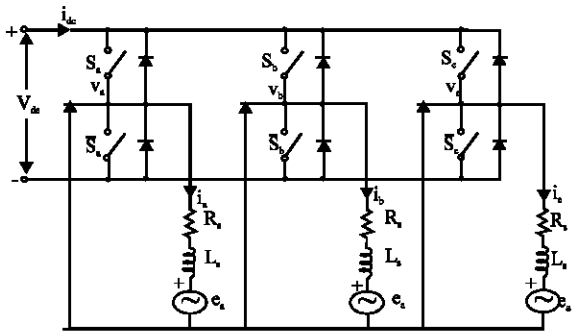

Fig. 2: Equivalent circuit of PMSM and inverter

The stator voltage equations, of PMSM in matrix form, can be represented as the statements of phase currents in (1). This equation can be shown in state-space form as in (2).

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} L_s & 0 & 0 \\ 0 & L_s & 0 \\ 0 & 0 & L_s \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} \quad (1)$$

$$\frac{d}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \begin{bmatrix} L_s & 0 & 0 \\ 0 & L_s & 0 \\ 0 & 0 & L_s \end{bmatrix}^{-1} \left\{ \begin{bmatrix} -R_s & 0 & 0 \\ 0 & -R_s & 0 \\ 0 & 0 & -R_s \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} - \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} + \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} \right\} \quad (2)$$

The rotor speed and electrical torque can be written as:

$$\frac{d}{dt}\omega_r = \frac{p}{2}\left(T_e - T_L - B\left(\frac{2}{p}\right)\omega_r \right) \Big/ J \quad (3)$$

$$T_e = K I^* \quad (4)$$

Where, K = -3p/4 λ_f and λ_f is the flux due to the permanent magnet rotor. The function of hysteresis band current controller is given in (5).

$$h_x = \begin{cases} 1 & \text{if } i_x^* - i_x \le 0.5 h_{rb} \\ 0 & \text{if } i_x^* - i_x \ge -0.5 h_{rb} \end{cases} \quad (5)$$

Where, x represents, respectively a, b and c. h_x represents the functions of hysteresis band current controller h_a,h_b and h_c. H_{rb} is the range of hysteresis band current controller. Using h_x, Eq. 2 can be represented by:

$$\frac{d}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \begin{bmatrix} L_s & 0 & 0 \\ 0 & L_s & 0 \\ 0 & 0 & L_s \end{bmatrix}^{-1} \left\{ \begin{bmatrix} -R_s & 0 & 0 \\ 0 & -R_s & 0 \\ 0 & 0 & -R_s \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} - \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} + \begin{bmatrix} \frac{(2h_a - h_b - h_c)}{3} \\ \frac{(-h_a + 2h_b - h_c)}{3} \\ \frac{(-h_a - h_b + 2h_c)}{3} \end{bmatrix} \begin{bmatrix} V_{dc} \end{bmatrix} \right\} \quad (6)$$

PARTICLE SWARM OPTIMIZATION (PSO)

The Particle Swarm Optimization Algorithm (PSOA) was firstly proposed by Eberhart and Kennedy (1995a, b) and has deserved some attention during the last years in the global optimization field. PSOA is based on the population of agents or particles and tries to simulate its social behavior in optimal exploration of problem space.

During time (iterations in the optimization context) each agent possesses a velocity vector that is a stochastic combination of its previous velocity and the distances of its current position to its own best ever position and to the best ever swarm position. The weights of the last two directions are controlled by two parameters called cognitive and social parameters (Bergh, 2001; Schutte and Groenwold, 2003).

PSOA belongs to a class of stochastic algorithms for global optimization and its main advantages are the easily parallelization and simplicity. PSOA seems to outperform the genetic algorithm for some difficult programming classes, namely the unconstrained global optimization problems (Bergh, 2001).

In spite of the referred advantages, PSOA possesses some drawbacks, namely its parameters dependency and the slow convergence rate in the vicinity of the global optimum.

PSOA is based on the population (swarm) of particles. Each particle is associated with velocity that indicates where the particle is *travelling*. Let t be a time instant. The new particle position is computed by adding the velocity vector to the current position:

$$x_p(t+1) = x_p(t) + v_p(t+1) \quad (7)$$

Where, $x_p(t)$ particle p's position, p = 1,..., s , at time instant t, $v_p(t+1)$ new velocity (at time t+1) and s is population size.
The velocity update equation is given by:

$$v_j^p(t+1) = t(t)v_j^p(t) + \mu\omega_{1j}(t)(y_j^p(t) - x_j^p(t)) + \nu\omega_{2j}(t)(y_j^g(t) - x_j^p(t)) \quad (8)$$

For j = 1,..., n , where $\tau(T)$ is a weighting factor (inertial),μ is the cognitive parameter and v is the social parameter. $\omega_{1j}(t)$ and $\omega_{2j}(t)$ are random numbers drawn from the uniform distribution U(0,1), used for each dimension j = 1,..., n . $y^p(t)$ is the best position previously visited by the p^{th} particle and $y^g(t)$ is the best position in the swarm found so far. PSO algorithm can be described as follows:

1. Randomly initialize the swarm positions $X=\{x^1(0)... x^s(0)\}$ and velocities $V = \{v^1(0)... v^s(0)\}$.
2. $t \leftarrow 0$ and $y^p(t) = x^p(t)$, p = 1,..., s.
3. For all p in {1... s} do:
   if $f(x^p(t)) < f(y^p(t))$ then $y^p(t+1) \leftarrow x^p(t)$
   else $y^p(t+1) \leftarrow y^p(t)$.
4. For all p in {1... s} do:
   Compute $v^p(t+1)$ and $x^p(t+1)$ using equations (7) and (8).
5. if the stopping criterion is true, then stop.
   else $t \leftarrow t+1$ and goes to step 3.

The stopping criterion mostly used in the literature is related to the function value at the global optimum. The algorithm stops, if either the objective function at the best ever particle is approximately equal to the known objective minimum, or a maximum number of iterations is exceeded.

THE IMPROVED PI CONTROLLER BASED ON PSO (PIPSO) FOR THE PMSM DRIVE

PMSM drive bloc scheme: The PIPSO controller for the PMSM drives is shown in Fig. 3. The PSO uses a population of agents which represent the controller parameters K_p and K_i.

Where, e_W represents the speed error and ω^* is the reference speed.

During the time step, each member of the swarm is evaluated on how well it minimizes the performance function given in (9).

$$F(K_p, K_i) = \alpha_1 \cdot e_w^2(k) + \alpha_2 \cdot (K_p \cdot e_w(k) + K_i \cdot e_w(k) \cdot T)^2 \quad (9)$$

α_1 and α_2 represent the importance weight of the first and the second terms of (9), respectively, T is the sample time, K_p and K_i are the gains of the PI controller.

In this application, feedback signals are the position θ and the phase currents. The position signal is used to calculate the speed. The switching signal generator is used to control turn-on angle θ_{on}, turn-off angle θ_{off} and pulse width modulation duty cycle.

Figure 4 shows that the PSO bloc receives the speed error e_W and provides the optimal parameters K_p and K_i for the PI bloc. This bloc exploits these parameters in order to generate the optimal reference currents i_{abcr}. Then, the currents loop, composed of a command rule and a 3 phase inverter, provides the optimal currents i_{abc} that will be used by the SM bloc to reach the required speed ω^*.

In this application, feedback signals are the position θ and the phase currents. The position signal is used to calculate the speed. The switching signal generator is used to control turn-on $angle_{on}$, turn-off $angle_{off}$ and pulse width modulation duty cycle.

Implementation of PSO to the PMSM drive: The PMSM used in this work implements a 3 phase permanent magnet synchronous machine with sinusoidal flux distribution. The sinusoidal machine is modeled in the dq rotor reference frame. Stator windings are connected in Wye to an internal neutral point.
The machine parameters are:

- Stator resistance: R_s= 2.875 Ω
- Inductance $L_d = L_q$= 8.5 e^{-3}H.
- Flux induced by magnets= 0.175wb.
- Inertia = 0.8 e^{-3}kg.m^2, friction factor= 0 N.m.s and pair of poles = 4.

The configuration of PSO parameters is given as follow:

Swarm size: The first step of PSO is to create the initial particles swarm. The swarm size used is 50. The positions and velocities of particles are represented by a real valued number.

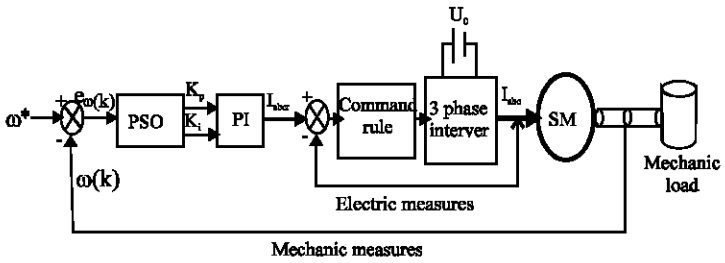

Fig. 3: The PIPSO controller for the PMSM drive

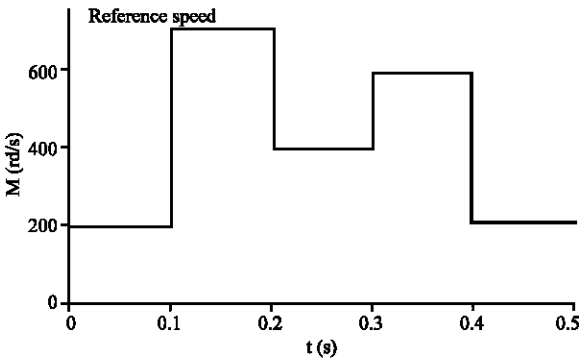

Fig. 4: Electric speed reference (Trapezoidal repeating sequence)

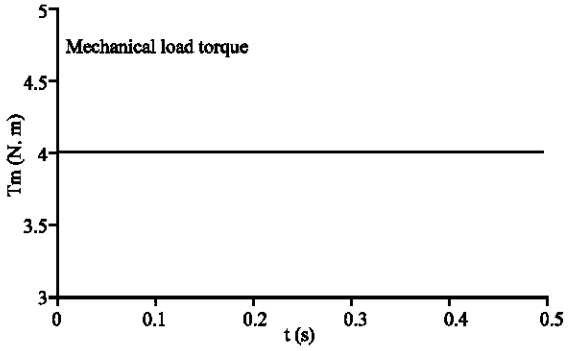

Fig. 5: Mechanical load torque (Tm = 4 N.m)

Variable bounds: The PSO algorithm is used to optimize the gains K_p and K_i of a PI controller. There are going to be two strings assigned to each position of the swarm, these members will be comprised of a string that will be evaluated throughout the evolution of the PSO. The range of the first parameter (K_p) is [50, 100] and the second one (K_i) belongs to (Slemon, 1994; Schutte and Graenwold, 2003).

- The weighting factor τ(T) used in this study is given as follow:

$$\tau(T)= 0.9-(t+1)*(0.5/(t+1))$$

- The cognitive parameter μ and the social parameter í are initialized at 2.

RESULTS AND DISCUSSION

Two studied cases are used. In the first case, the reference electric speed is defined by a trapezoidal repeating sequence (Fig. 5) with a constant mechanical load torque (Tm = 4Nm) (Fig. 6). In the second case, the reference electric speed and the mechanical load torque are defined by a step as depicted in Fig. 7 and 8, respectively.

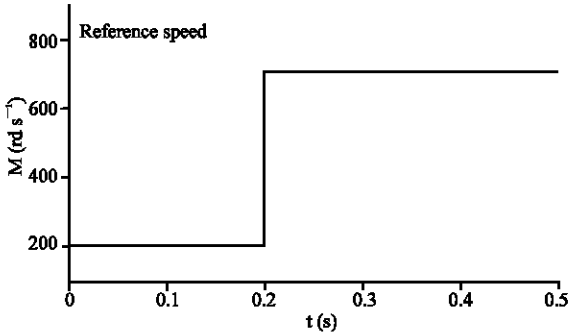

Fig. 6: Electric speed reference (step)

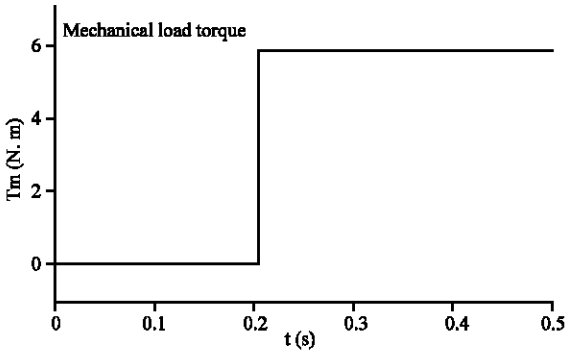

Fig. 7: Mechanical load torque (step)

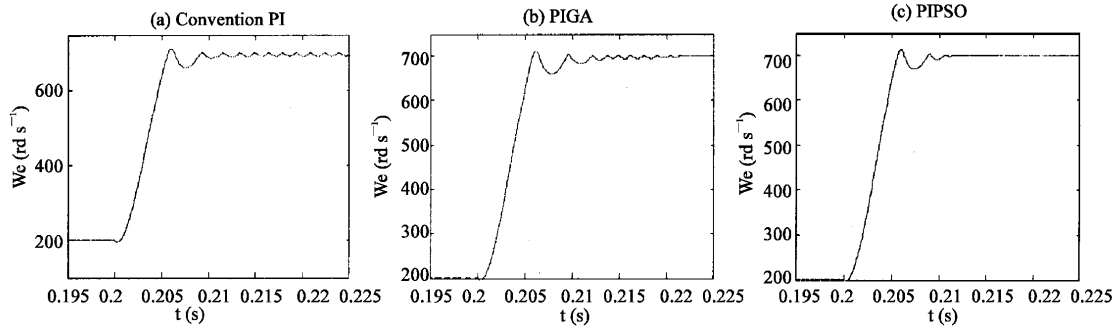

Fig. 8: Electric speed response

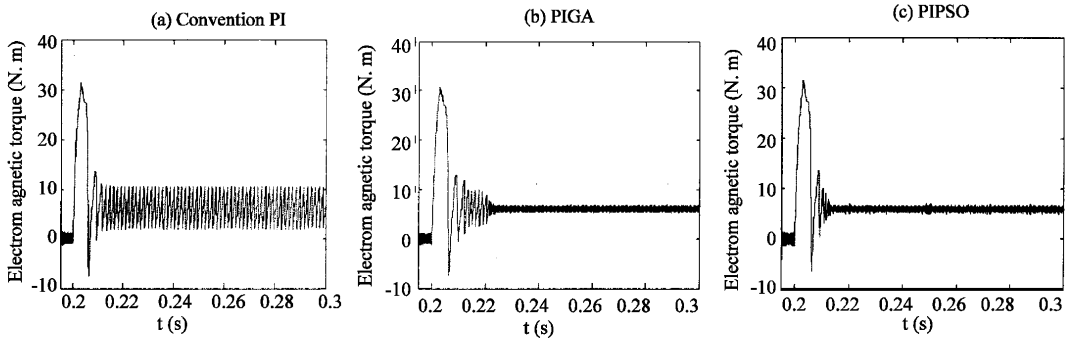

Fig. 9: Electromagnetic torque response

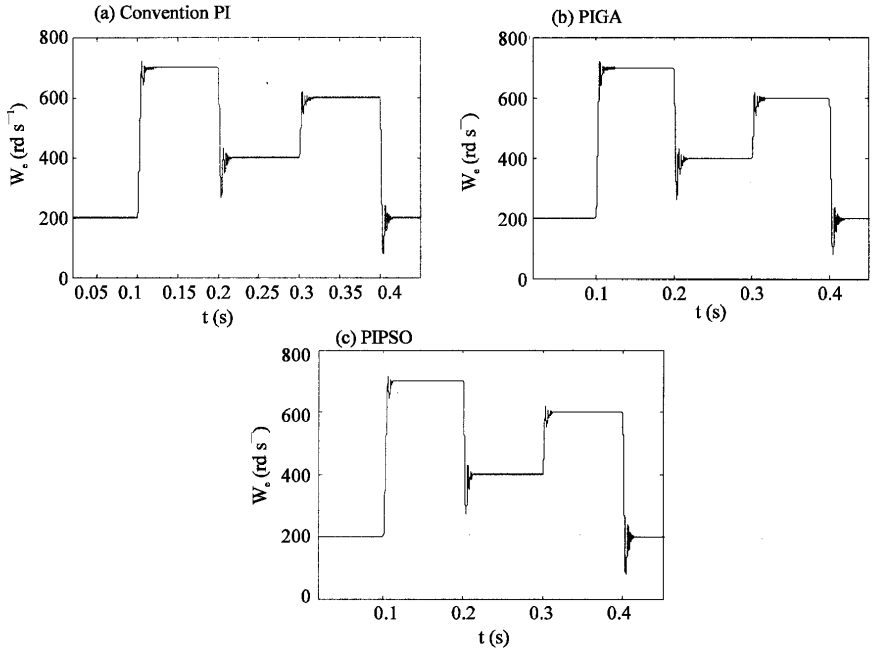

Fig.10: Electric speed response

Case 1: In this studied case, the reference electric speed and the mechanical load torque are defined by a step as depicted in Fig. 6 and 7, respectively.

Figure 8 presents the simulation results related to the three control strategies, i.e., conventional PI (Fig. 9a), PIGA (Fig. 8b) and PIPSO (Fig. 8c). These results display the electric speed response time.

Figure 8a shows that the electric speed response is not reached for the conventional PI. However, the electric speed response time related to PIGA, is t ˜ 0.222s

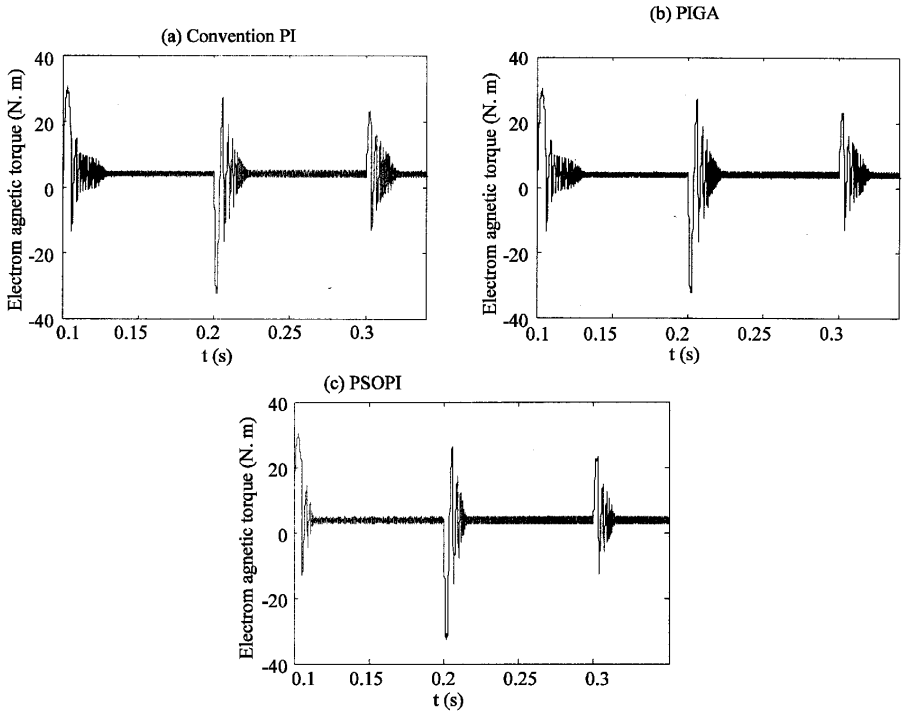

Fig. 11: Electromagnetic torque response

(Fig. 8b) while at t ˜ 0.21s, the speed response time is reached using PIPSO (Fig. 9c). Hence, the PIPSO is 94% faster than PIGA.

It is obvious that the performance of PIPSO in electric speed response is better than both controller strategies, i.e., conventional and PIGA, in this case.

In addition, to illustrate the performance and the efficiency of the proposed controller, Fig. 10 presents the electromagnetic torque response provided by the three controllers.

Electromagnetic torque response depicted in Fig. 9a, provided by the conventional controller, shows that oscillations are not reduced during the simulation time. In Fig. 10b, oscillations are reduced at t ˜0.221s according to the use of PIGA (Fig. 9b) whereas with PIPSO, oscillations are attenuated at 0.218s (Fig. 9c). In this case, the time rate PIGA / PIPSO is 101.4%.

Case 2: The reference electric speed is defined by a trapezoidal repeating sequence (Fig. 4) with a mechanical load torque constant (Tm = 4Nm) (Fig. 5).

Figure 10 presents the simulation results related to the three control strategies, i.e., conventional PI (Fig. 10a), PIGA (Fig. 10b) and PIPSO (Fig. 10c). These results display the electric speed response time.

As shown in Fig. 10, the PIPSO is more appropriate than both controllers (conventional and PIGA), at the different phases of the MSAP control, in terms of stability and response time.

Figure 11 illustrates the electromagnetic torque response. This figure confirms the results mentioned above.

CONCLUSION

In this study the speed control problem of Permanent magnet synchronous drive is studied. The nonlinear behaviour of the system, non-matched perturbations, parameter variations and load torque disturbance limit the performances of classical linear controllers used for this purpose. In fact, using the conventional PI, convergence is occasionally reached and depends generally on a fine tuning of parameters, therefore, this controller is not efficient in real time.

To surmount this problem, a new control strategy based on particle swarm (PIPSO) is proposed. As an intelligent control technology, this proposed model provides robust adaptive response for a drive with nonlinearity, parameter variation and load disturbance effect.

From the simulation results, it is clear that the PIPSO controller provides better speed response and accuracy over the genetic and conventional PI controllers.

This work validates the adaptation of PSO to PMSM drive with low power. It will be interesting to extend it for alternative current machines (synchronous and asynchronous) with high power.

REFERENCES

Akcayol, M.A., A. Cetin and C. Elmas, 2002. An Educational Tool for Fuzzy Logic Controlled BDCM. IEEE. Trans. Edu., 45: 33-42.

Bergh, van den, F., 2001. An Analysis of Particle Swarm Optimizers. PhD thesis, Faculty of Natural and Agricultural Science, University of Pretoria.

Eberhart, R. and J. Kennedy, 1995. New optimizers using particle swarm theory. In: Proc. 6th Int. Sysmp. Micro Machine and Human Sci., pp: 39-43.

Karunadasa, J.P. and A.C. Renfrew, 1991. Design and implementation of microprocessor based sliding mode controller for brushless servomotor. IEEE. Proc. B., 138: 345-363.

Kennedy, J. and R. Eberhart, 1995. Particle swarm optimization. In: Proc. IEEE Int. Conf. Neural Networks, Perth, Australia. IEEE. Service Center, Piscataway, NJ. http://engr.iupui.edu/~shi/Coference pso pap4.html, pp: 1942.1948.

Lee, C.C., 1990. Fuzzy logic in control systems: Fuzzy logic controller, Part I, Part II. IEEE. Trans. Sys., Man and Cyber., 20: 404-435.

Lin, C.T. and C.S.G. Lee, 1991. Neural-network-based Fuzzy logic control and decision system. IEEE. Trans. Comp., 40: 1320-1336.

Rahman, M.A. and M.A. Hoque, 1998. On-line adaptive artificial neural network based vector control of permanent magnet synchronous motors. IEEE. Trans. Energy Conversion, 13: 311-318.

Schutte, J.F. and A.A. Groenwold, 2003. A study of global optimization using particle swarms. J. Global Optimization, 31: 93-108.

Slemon, G.R., 1994. Electrical machines for variable-frequency drives. Proc. IEEE., 82: 1123-1139.